Robust Tensor Factorization with Unknown Noise

Xi'ai Chen[1,2], Zhi Han[1*], Yao Wang[1,3], Qian Zhao[3], Deyu Meng[3], Yandong Tang[1]
[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences;
[2]University of Chinese Academy of Sciences; [3]Xi'an Jiaotong University
{chenxiai, hanzhi, ytang}@sia.cn, {yao.s.wang, timmy.zhaoqian}@gmail.com, dymeng@mail.xjtu.edu.cn

Abstract

Because of the limitations of matrix factorization, such as losing spatial structure information, the concept of tensor factorization has been applied for the recovery of a low dimensional subspace from high dimensional visual data. Generally, the recovery is achieved by minimizing the loss function between the observed data and the factorization representation. Under different assumptions of the noise distribution, the loss functions are in various forms, like L_1 and L_2 norms. However, real data are often corrupted by noise with an unknown distribution. Then any specific form of loss function for one specific kind of noise often fails to tackle such real data with unknown noise. In this paper, we propose a tensor factorization algorithm to model the noise as a Mixture of Gaussians (MoG). As MoG has the ability of universally approximating any hybrids of continuous distributions, our algorithm can effectively recover the low dimensional subspace from various forms of noisy observations. The parameters of MoG are estimated under the EM framework and through a new developed algorithm of weighted low-rank tensor factorization (WLRTF). The effectiveness of our algorithm are substantiated by extensive experiments on both of synthetic data and real image data.

1. Introduction

The problem of recovering a low dimensional linear subspace from high dimensional visual data naturally arises in the fields of computer vision, machine learning and statistics, and has drawn increasing attention in the recent years. Typical examples include representation and recognition of faces [25, 20, 22, 1], structure from motion [21], recognition of 3D objects under varying pose [15], motion segmentation [23].In such contexts, the data to be analyzed usually can be formulated as high-order tensors, which are natural generalization of vectors and matrices. Existing approaches, including LRMF and RPCA, proceed by unfolding tensors into matrices and then applying common matrix techniques to deal with tensor problems. However, as shown in [10], such matricization fails to exploit the essential tensor structure and often leads to suboptimal procedure. Figure 1 illustrates the difference between the matrix based method and tensor based method in dealing with the high-order tensor data. The upper row is the matrix based factorization method, which needs to preliminarily unfold or vectorize the tensor; the lower row is the tensor based method which directly factorize the tensor without destroying the spatial structures. Given a high-order tensor data, an efficient way to extract the underlying useful information is low-rank tensor factorization (LRTF), which aims to extract low-rank subspaces underlying those vector spaces so that the original tensor can be suitably expressed through reasonably affiliating these subspaces. In the recent years, the application of LRTF has been extended to a wide range of fields throughout science and engineering [7].

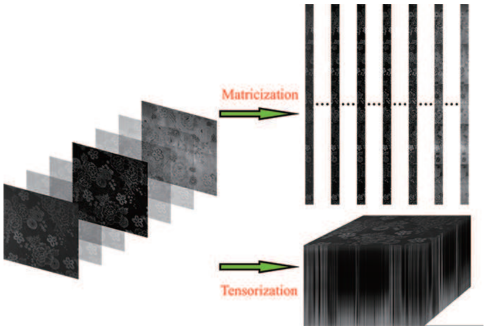


Figure 1. High-order data represented by tensorization better preserves the essential data structure compared with matricizaiton.

The notation in this paper are defined as follows. Scalars are denoted by lowercase letters $(a, b, ...)$ and vectors are denoted by bold lowercase letters $(\mathbf{a}, \mathbf{b}, ...)$ with elements $(a_i, b_j, ...)$. Matrices are represented by uppercase letters $(A, B, ...)$ with column vectors $(\mathbf{a}_{:j}, \mathbf{b}_{:j}, ...)$ and elements $(a_{ij}, b_{ij}, ...)$. The calligraphic letters $(\mathcal{A}, \mathcal{B}, ...)$ stand for the the high-order tensors. A K-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ is a rank-1 tensor, if it can be written as the outer product of K vectors, i.e., $\mathcal{X} = \mathbf{a}^1 \circ \mathbf{a}^2 \circ \cdots \circ \mathbf{a}^K$. Then

[*]Corresponding author.

the element of the tensor can be represented as: $x_{i_1 i_2 \cdots i_K} = a^1_{i_1} a^2_{i_2} \cdots a^K_{i_K}$. The slice of a K-order tensor is a matrix defined by fixing every index but two. Therefore the slice of a 3-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ has the form: frontal slices $X_{::k}$, lateral slices $X_{:j:}$, horizontal slices $X_{i::}$.

As discussed in [7], although there are several tensor factorization forms, our framework for LRTF is based on the CANDECOMP/PARAFAC (CP) decomposition. The main reason is that the CP decomposition can be viewed as a higher-order generalization of the matrix singular value decomposition [2] and has been widely used in many real applications [31, 26, 17]. Mathematically, a K-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, with the integer I_k $(1 \leq k \leq K)$ indicating the dimension of $\mathcal{X}$ along the k-th order, is represented in the CP decomposition form as:

$$\mathcal{X} = \sum_{d=1}^{r} \mathbf{u}^d \circ \mathbf{v}^d \circ \cdots \circ \mathbf{t}^d, \tag{1}$$

where r is assumed to be the rank of the tensor $\mathcal{X}$. Then each element of the tensor has the following form:

$$x_{ij\ldots k} = \sum_{d=1}^{r} u_i^d v_j^d \cdots t_k^d. \tag{2}$$

It is known that the canonical fit function for the CP LRTF is based on the Frobenius norm function which assumes the noise to follow a Gaussian distribution. However, for many real data, such as the fMRI neuroimaging data [5] and the video surveillance data [8], a relative large perturbation in magnitude only affects a relatively small fraction of data points, which often violates the Gaussian assumption and instead follows a Laplacian distribution.

Therefore, it is necessary to consider other loss function that is robust to Laplacian noise. To alleviate this problem, one commonly used strategy is to replace the Frobenius norm function (say, L_F norm) by the L_1-type norm [6, 3], which is known to be robust to gross Laplacian perturbations. Unfortunately, in many real applications, the noise often exhibits very complex statistical distributions rather than a single purely Gaussian or Laplacian noise [29]. This motivates us to consider more flexible modeling strategies to tackle such complex noise cases.

Under the framework of low-rank matrix factorization (LRMF), Meng and De la Torre [13] firstly proposed to model the noise as Mixture of Gaussians (MoG). They showed that the MoG model is a universal approximator to any continuous distribution, and hence could be capable of modeling a wider range of noise distributions. Along this line, Zhao et al. [30] further extended the MoG model to deal with robust PCA (RPCA) problem. Extensive experiments on synthetic data, face modeling and background subtraction demonstrated the merits of MoG model.

As such, to share the same light of matrix MoG model, we aim to introduce a novel MoG model to the tensor case for the LRTF task to overcome the drawbacks of existing models, which only model one simple Gaussian or Laplacian noise.

The contributions of this paper can be summarized as follows: (1) We propose a new low-rank subspace learning approach called weighted low-rank tensor factorization (WLRTF), which preserves the essential tensor structure; (2) We apply MoG to the proposed WLRTF called weighted low-rank tensor factorization based on MoG (MoG WLRTF); (3) For solving the proposed model, we propose efficient algorithms to estimate the parameters under the EM framework and through the proposed algorithm of WLRTF. Our strategy is different from not only the traditional EM algorithm for solving matrix/tensor decomposition models, but also conventional alternative least squares (ALS) techniques for solving other tensor decomposition problems. A series of synthetic and real data experiments are then provided to validate the effectiveness of our method. The source codes of our algorithm are published online: *http://vision.sia.cn/our%20team/Hanzhi-homepage/vision-ZhiHan(English).html.*

2. Weighted low-rank tensor factorization based on MoG

In this section, a new tensor model for modeling complex noise is proposed. Firstly, MoG is applied to model the noise element of the input tensor and thus have the log-likelihood optimization objective. Then through assuming a latent variable with higher dimension, we solve the problem iteratively under the EM framework. Finally, based on CP decomposition, we design a new algorithm that is different from ALS to solve the weighted low-rank tensor factorization in order to update each factorized tensor component.

2.1. CP decomposition with MoG

Taking the noise part (denoted as ε_{ijk}) into consideration, each element x_{ijk} $(i = 1, 2, ..., I, j = 1, 2, ..., J, k = 1, 2, ..., K)$ of the 3-order tensor $\mathcal{X}$ in CP decomposition can be written as:

$$x_{ijk} = \sum_{d=1}^{r} u_i^d v_j^d t_k^d + \varepsilon_{ijk}. \tag{3}$$

As MoG has the ability to universally approximate any hybrids of continuous distributions, it is adopted for modeling the unknown noise in the original data. Hence every ε_{ijk} follows an MoG and the distribution $p(\varepsilon)$ is defined as:

$$p(\varepsilon) \sim \sum_{n=1}^{N} \pi_n \mathcal{N}(\varepsilon | \mu_n, \sigma_n^2), \tag{4}$$

where π_n is the mixing proportion with $\pi_n \geq 0$ and $\sum_{n=1}^{N} \pi_n = 1$. $\mathcal{N}(\varepsilon|\mu_n, \sigma_n^2)$ denotes the Gaussian distribution with mean μ_n and variance σ_n^2.

Then every x_{ijk} in Eq. (3) follows a MoG distribution with mean $\Lambda_n = \sum_{d=1}^{r} u_i^d v_j^d t_k^d + \mu_n$ and variance σ_n^2. The probability of each element x_{ijk} in the input tensor $\mathcal{X}$ can thus be represented as:

$$p(x_{ijk} \mid \Pi, \Lambda, \Sigma) = \sum_{n=1}^{N} \pi_n \mathcal{N}(x_{ijk} \,|\Lambda_n,\ \sigma_n^2), \qquad (5)$$

where $\Pi = \{\pi_1, \pi_2, ..., \pi_N\}, \Lambda = \{\Lambda_1, \Lambda_2, ..., \Lambda_n\}, \Sigma = \{\sigma_1, \sigma_2, ..., \sigma_N\}$.

We then define the likelihood of $\mathcal{X}$ as

$$p(\mathcal{X}\,|\Pi, \Lambda, \Sigma) = \prod_{i,j,k\in\Omega} \sum_{n=1}^{N} \pi_n \mathcal{N}(x_{ijk} \,|\Lambda_n,\ \sigma_n^2), \qquad (6)$$

where Ω is the index set of the non-missing entries of $\mathcal{X}$.

The goal is to maximize the log-likelihood function with respect to the parameters Π, Λ, Σ, i.e.

$$\max_{\Pi,\Lambda,\Sigma} \mathcal{L}(\Pi, \Lambda, \Sigma) = \sum_{i,j,k\in\Omega} \log \sum_{n=1}^{N} \pi_n \mathcal{N}(x_{ijk} \,|\Lambda_n,\ \sigma_n^2). \qquad (7)$$

2.2. EM algorithm

EM algorithm [4] is proven to be effective for solving the maximization problem of the log-likelihood function. Therefore, for solving Eq. (7), we assume a higher dimensional latent variable under the EM framework. Then the original problem can be viewed as a Gaussian Scale Mixtures (GSM) with μ_n assumed to be 0, which has been widely used in previous works [19, 24]. Define $U = \{\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_r}\}, V = \{\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_r}\}, T = \{\mathbf{t_1}, \mathbf{t_2}, ..., \mathbf{t_r}\}$, and then Eq. (7) can be rewritten as:

$$\begin{aligned} &\max_{U,V,T,\Pi,\Sigma} \mathcal{L}(U, V, T, \Pi, \Sigma) \\ &= \sum_{i,j,k\in\Omega} \log \sum_{n=1}^{N} \pi_n \mathcal{N}(x_{ijk} \left| \sum_{d=1}^{r} u_{id} v_{jd} t_{kd},\ \sigma_n^2). \right. \end{aligned} \qquad (8)$$

In the model, the variables U, V, T are shared by all the clusters of MoG and the mean for each cluster of the standard EM algorithm is represented by them. Thus our proposed algorithm will iterate between computing responsibilities of all Gaussian components (**E Step**) and maximizing the parameters Π, Σ and U, V, T in the model (**M Step**).

***E Step*:** A latent variable z_{ijkn} is assumed in the model, with $z_{ijkn} \in \{0, 1\}$ and $\sum_{n=1}^{N} z_{ijkn} = 1$, representing the assigned value of the noise ε_{ijk} to each component of the mixture. Here we denote $Z = \{z_{ijkn} | i = 1, 2, ..., I; j = 1, 2, ..., J; k = 1, 2, ..., K; n = 1, 2, ..., N\}$. The posterior responsibility of the n-th mixture for generating the noise of x_{ijk} can be calculated by

$$\begin{aligned} &E(z_{ijkn}) = \gamma_{ijkn} \\ &= \frac{\pi_n \mathcal{N}(x_{ijk} \left| \sum_{d=1}^{r} u_{id} v_{jd} t_{kd},\ \sigma_n^2) \right.}{\sum_{n=1}^{N} \pi_n \mathcal{N}(x_{ijk} \left| \sum_{d=1}^{r} u_{id} v_{jd} t_{kd},\ \sigma_n^2) \right.}. \end{aligned} \qquad (9)$$

The **M step** maximizes the upper bound given by the **E step** with regard to U, V, T, Π, Σ:

$$\begin{aligned} &E_Z p(\mathcal{X}, Z|U, V, T, \Pi, \Sigma) = \sum_{i,j,k\in\Omega} \sum_{n=1}^{N} \gamma_{ijkn}(log\pi_n \\ &- log\sqrt{2\pi\sigma_n} - \frac{(x_{ijk} - \sum_{d=1}^{r} u_{id} v_{jd} t_{kd})^2}{2\pi\sigma_n^2}). \end{aligned} \qquad (10)$$

This maximization problem can be solved by alternatively updating the MoG parameters Π, Σ and the factorized matrices U, V, T as follows:

***M Step** to update* Π, Σ: The closed-form updates for the MoG parameters are:

$$\begin{aligned} &m_n = \sum_{i,j,k} \gamma_{ijkn}, \pi_n = \frac{m_n}{\sum_{n} m_n}, \\ &\sigma_n^2 = \frac{1}{m_n} \sum_{i,j,k} \gamma_{ijkn}(x_{ijk} - \sum_{d=1}^{r} u_{id} v_{jd} t_{kd})^2. \end{aligned} \qquad (11)$$

***M Step** to update U, V, T*: Re-write Eq. (10) only with regard to the unknown components U, V, T as follows:

$$\begin{aligned} &\sum_{i,j,k\in\Omega} \sum_{n=1}^{N} \gamma_{ijkn}(-\frac{(x_{ijk} - \sum_{d=1}^{r} u_{id} v_{jd} t_{kd})^2}{2\pi\sigma_n^2}) \\ &= -\sum_{i,j,k\in\Omega} \sum_{n=1}^{N} (\frac{\gamma_{ijkn}}{2\pi\sigma_n^2})(x_{ijk} - \sum_{d=1}^{r} u_{id} v_{jd} t_{kd})^2 \\ &= -\left\| \mathcal{W} \odot (\mathcal{X} - \sum_{d=1}^{r} \mathbf{u}_{:d} \circ \mathbf{v}_{:d} \circ \mathbf{t}_{:d}) \right\|_{L_F}^2. \end{aligned} \qquad (12)$$

Here $\odot$ denotes the Hadamard product (component-wise multiplication) and the element w_{ijk} of $\mathcal{W} \in \mathbb{R}^{I \times J \times K}$ is

$$w_{ijk} = \begin{cases} \sqrt{\sum_{n=1}^{N} \frac{\gamma_{ijkn}}{2\pi\sigma_n^2}}, & i, j, k \in \Omega \\ 0, \quad i, j, k \notin \Omega. \end{cases} \qquad (13)$$

The whole MoG WLRTF optimization process is summarized in Algorithm 1. Note that in the **M Step**, U, V and T are evaluated by solving the WLRTF model $\min\limits_{U,V,T} \left\| \mathcal{W} \odot (\mathcal{X} - \sum\limits_{d=1}^{r} \mathbf{u}_{:d} \circ \mathbf{v}_{:d} \circ \mathbf{t}_{:d}) \right\|_{L_F}^2$, which will be introduced in details in the following section.

Algorithm 1 (EM algorithm for MoG WLRTF)

```
Input: $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, each image size is $I \times J$ and the number of images is $K$.
Output: $U, V, T$
Initialize $\Pi, \Sigma, U, V, T$, MoG number N, small threshold $\epsilon$.
while not converged do
    E Step:
    Evaluate $\gamma_{ijkn}$ for $i = 1, 2, ...I; j = 1, 2, ..., J; k = 1, 2, ..., K; n = 1, 2, ..., N$ by Eq. (9).
    M Step for $\Pi, \Sigma$:
    Evaluate $\pi_n, \sigma_n^2$ for $n = 1, 2, ..., N$ by Eq. (11)
    M Step for $U, V, T$:
    Evaluate $U, V, T$ by solving
    $\min\limits_{U,V,T} \left\| \mathcal{W} \odot (\mathcal{X} - \sum\limits_{d=1}^{r} \mathbf{u}_{:d} \circ \mathbf{v}_{:d} \circ \mathbf{t}_{:d}) \right\|_{L_F}^2$,
    where $\mathcal{W}$ is calculated by Eq. (13).
end while
```

2.3. Weighted low-rank tensor factorization

The WLRTF error model of the three-dimensional tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is written as

$$\min_{U,V,T} \left\| \mathcal{W} \odot (\mathcal{X} - \sum_{d=1}^{r} \mathbf{u}_{:d} \circ \mathbf{v}_{:d} \circ \mathbf{t}_{:d}) \right\|_{L_F}, \tag{14}$$

where $U \in \mathbb{R}^{I \times r}, V \in \mathbb{R}^{J \times r}, T \in \mathbb{R}^{K \times r}$ are low-dimensional matrix with rank r. $\mathcal{W} \in \mathbb{R}^{I \times J \times K}$ is the weighted tensor which is composed by the standard variance of the input tensor elements.

Because of the effectiveness and implementation convenience of ALS, we adopt its idea to update U, V, T of the tensor one at a time.

Suppose $\mathbf{I_1^0}, ..., \mathbf{I_n^0} \in \mathbb{R}^{w \times h}$ are data matrices. In order to stack each of the above matrix as a vector, we define the operator $vec : \mathbb{R}^{\mathrm{w} \times h} \to \mathbb{R}^{wh}$.

For each slice of the higher-order tensor, it can be viewed as a linear combination of the corresponding slices of all the rank-1 tensors. Different from other methods for solving the problem of LRTF, we stack each frontal slice of the higher-order tensor as a vector of a new matrix denoted as M_F. Correspondingly, the vectorized horizontal slices and lateral slices are represented as M_H and M_L, respectively.

Firstly we have

$$\mathcal{X}^{new} = \mathcal{W} \odot \mathcal{X}. \tag{15}$$

Then taking term T as an example, the vectorized frontal slice M_F of the higher-order tensor can be written as follows:

$$M_F = [vec(X_{::1}^{new})|...|vec(X_{::K}^{new})] \in \mathbb{R}^{IJ \times K}. \tag{16}$$

For the i-th frontal slice of the higher-order tensor, the vectorized corresponding slices of all the rank-1 tensors can be viewed as the i-th element of the cell F which can be represented as:

$$\begin{aligned} F_i = [&vec(W_{::i} \odot (\mathbf{u}_{:1} \circ \mathbf{v}_{:1}))|... \\ &|vec(W_{::i} \odot (\mathbf{u}_{:r} \circ \mathbf{v}_{:r}))] \in \mathbb{R}^{IJ \times r}. \end{aligned} \tag{17}$$

Then the i-th vector of term T can be updated as follows:

$$T_{i:} = (F_i^{\dagger} M_{F:i})^{\mathrm{T}} \in \mathbb{R}^{1 \times r}, \tag{18}$$

where $A^{\dagger}$ represents the pseudo-inverse matrix of matrix A, and B^{T} denotes the transposed matrix of matrix B.

Similarly, we have the term V and U updated as following:

$$M_L = [vec(X_{:1:}^{new})|...|vec(X_{:J:}^{new})] \in \mathbb{R}^{IK \times J}, \tag{19}$$

$$\begin{aligned} L_i = [&vec(W_{:i:} \odot (\mathbf{t}_{:1} \circ \mathbf{u}_{:1}))|... \\ &|vec(W_{:i:} \odot (\mathbf{t}_{:r} \circ \mathbf{u}_{:r}))] \in \mathbb{R}^{IK \times r}, \end{aligned} \tag{20}$$

$$V_{i:} = (L_i^{\dagger} M_{L:i})^{\mathrm{T}} \in \mathbb{R}^{1 \times r}. \tag{21}$$

$$M_H = [vec(X_{1::}^{new})|...|vec(X_{I::}^{new})] \in \mathbb{R}^{JK \times I}, \tag{22}$$

$$\begin{aligned} H_i = [&vec(W_{i::} \odot (\mathbf{v}_{:1} \circ \mathbf{t}_{:1}))|... \\ &|vec(W_{i::} \odot (\mathbf{v}_{:r} \circ \mathbf{t}_{:r}))] \in \mathbb{R}^{JK \times r}, \end{aligned} \tag{23}$$

$$U_{i:} = (H_i^{\dagger} M_{H:i})^{\mathrm{T}} \in \mathbb{R}^{1 \times r}. \tag{24}$$

The WLRTF optimization process is summarized in Algorithm 2.

Algorithm 2 (WLRTF)

```
Input: The input tensor $\mathcal{X}$, initialized tensor factors $U, V, T$, weighted tensor $\mathcal{W}$, number of iteration and the threshold $\epsilon$.
Output: $U, V, T$.
while not converged do
    update $T$ with Eq. (16), (17), (18);
    update $V$ with Eq. (19), (20), (21);
    update $U$ with Eq. (22), (23), (24).
end while
```

Table 1. Predictive performance of competing methods with varied missing rate.

		MC ALM	MoG LRMF	HaLRTC	BM4D	LRTA	PARAFAC	MSI DL	CWM LRTF	WLRTF	MoG WLRTF
20%	E1	7.37	0.08	2.55e+02	7.67e+02	4.40e+02	4.38e+02	2.98e+02	3.51e+02	8.79e-05	**1.61e-08**
	E2	5.40	1.08e-04	6.50e+04	1.47e+03	6.47e+02	6.47e+02	3.70e+02	3.93e+02	2.11e-11	**6.82e-19**
	E3	7.25e+02	0.09	2.68e+04	9.69e+02	6.07e+02	5.95e+02	4.10e+02	4.63e+02	1.20e-04	**2.03e-08**
	E4	2.96e+02	0.57	3.76e+06	1.90e+03	9.62e+02	9.48e+02	5.82e+02	5.34e+02	3.69e-11	**8.85e-19**
40%	E1	8.28	1.24	2.55e+02	7.77e+02	5.28e+02	5.30e+02	3.58e+02	4.38e+02	0.25	**8.84e-09**
	E2	10.9	0.02	6.50e+04	2.06e+03	1.24e+03	1.23e+03	7.95e+02	8.07e+02	2.10e-04	**3.18e-19**
	E3	1.82e+03	7.36e+02	5.20e+04	1.29e+03	9.93e+02	9.78e+02	6.54e+02	8.00e+02	0.51	**1.90e-08**
	E4	6.08e+02	1.41e+02	7.06e+06	3.23e+03	2.28e+03	2.22e+03	1.50e+03	1.56e+03	5.87e-04	**9.63e-19**
60%	E1	40.8	5.01	2.55e+02	8.31e+02	7.11e+02	6.63e+02	4.85e+02	5.57e+02	6.93	**2.49e-07**
	E2	90.2	0.66	6.50e+04	2.99e+03	2.34e+03	2.21e+03	1.67e+03	1.90e+03	0.21	**2.77e-16**
	E3	2.86e+03	1.51e+04	7.89e+04	2.03e+03	1.86e+03	1.81e+03	1.25e+03	1.73e+03	34.6	**1.12e-06**
	E4	8.99e+02	1.57e+03	1.09e+07	7.20e+03	6.26e+03	6.21e+03	4.32e+03	6.22e+03	5.42	**4.21e-15**

Table 2. Reconstruction performance of competing methods with unknown noise.

		MC ALM	MoG LRMF	HaLRTC	BM4D	LRTA	PARAFAC	MSI DL	CWM LRTF	WLRTF	MoG WLRTF
Gaussian Noise	E1	**6.36**	32.3	2.55e+02	1.00e+03	6.31e+02	6.41e+02	4.42e+02	4.69e+02	59.4	54.2
	E2	11.9	**2.44**	6.50e+04	2.74e+03	1.55e+03	1.57e+03	1.17e+03	7.04e+02	7.00	5.84
	E3	8.62e+02	**11.4**	1.05e+05	1.27e+03	8.59e+02	8.53e+02	6.02e+02	6.09e+02	32.2	29.2
	E4	3.57e+02	72.0	1.41e+07	3.38e+03	2.10e+03	2.07e+03	1.53e+03	8.81e+02	1.77	**1.52**
Sparse Noise	E1	**15.5**	4.20e+02	5.10e+02	1.15e+03	1.05e+03	9.63e+02	7.70e+02	8.86e+02	7.00e+02	6.93e+02
	E2	**24.3**	4.96e+02	1.30e+05	3.47e+03	2.96e+03	2.63e+03	2.24e+03	2.47e+03	1.30e+03	1.42e+03
	E3	1.87e+03	5.25e+03	1.02e+05	1.12e+03	1.02e+03	1.05e+03	7.47e+02	8.41e+02	7.08e+02	**5.10e+02**
	E4	6.70e+02	1.04e+03	1.33e+07	2.64e+03	2.17e+03	2.34e+03	1.64e+03	1.77e+03	9.22e+02	**4.33e+02**
Mixture Noise	E1	**17.4**	4.63e+02	5.10e+02	1.39e+03	1.19e+03	1.15e+03	8.17e+02	1.07e+03	7.35e+02	6.68e+02
	E2	**26.3**	6.05e+02	1.30e+05	4.77e+03	3.90e+03	3.70e+03	2.58e+03	3.31e+03	1.43e+03	1.37e+03
	E3	1.99e+03	1.23e+04	1.06e+05	1.34e+03	1.13e+03	1.17e+03	7.85e+02	1.10e+03	6.45e+02	**4.59e+02**
	E4	7.15e+02	1.46e+03	1.50e+07	3.69e+03	2.84e+03	2.94e+03	1.93e+03	2.91e+03	7.70e+02	**3.83e+02**

3. Experiments

In this section, we conduct extensive experiments on both synthetic data and real applications to validate the effectiveness of the proposed MoG WLRTF algorithm compared with MC ALM [9], MoG LRMF [13], HaLRTC [10], BM4D [12], LRTA [18], PARAFAC [11], MSI DL [16], CWM LRTF [14] and our proposed tensor factorization algorithm WLRTF without MoG. For the matrix based methods, the tensor is firstly unfolded into matrix structure before processing. The synthetic experiments are designed to quantitatively assess our method from: i) predictive performance over missing entries given an incomplete tensor; ii) reconstruction performance given a both incomplete and noisy tensor. The three real data applications are image inpainting, multispectral image recovery and real hyperspectral image restoration for evaluating the robust completion performance.

3.1. Synthetic Experiments

The synthetic tensor is generated as follows: firstly, matrices $\{U, V, T\}$ are drawn from a standard normal distribution, i.e., $\forall i, j, k$, the vectors $\mathbf{u}_i, \mathbf{v}_j, \mathbf{t}_k$ of the matrices $\{U, V, T\}$ comply with a standard normal distribution $\mathcal{N}(0, I_{\mathbb{R}})$; Secondly, construct the true tensor by $\mathcal{X}_{gt} = [\![U, V, T]\!]$, and set the size to $10 \times 10 \times 10$ and CP rank $r = 5$. Then we conduct two synthetic experiments: i) for validating the predictive performance, we vary the true tensor missing entries rate (20%, 40%, 60%) ; ii) for verifying the reconstruction performance, we randomly choose 20% missing entries of the true tensor and further add certain type of noise to it as the following procedure: (1) Gaussian noise $\mathcal{N}(0, 0.1)$; (2) Sparse noise: 20% of the non-missing entries with the uniformly distribution over [-5,5]; (3) Mixture noise: 20% of the non-missing elements with the uniformly distribution over [-5,5], and 20% of the rest non-missing with Gaussian noise $\mathcal{N}(0, 0.2)$ and the rest with $\mathcal{N}(0, 0.01)$. The performance of each method is quantitatively assessed by the following measurements as used in [13]:

$$E1 = \|\mathcal{W} \odot (\mathcal{X}_{no} - \mathcal{X}_{rec})\|_{L_1}, E2 = \|\mathcal{W} \odot (\mathcal{X}_{no} - \mathcal{X}_{rec})\|_{L_2}$$
$$E3 = \|\mathcal{X}_{gt} - \mathcal{X}_{rec}\|_{L_1}, E4 = \|\mathcal{X}_{gt} - \mathcal{X}_{rec}\|_{L_2},$$

where $\mathcal{X}_{no}$ and $\mathcal{X}_{rec}$ are used to denote the noisy tensor and the recovered tensor, respectively. As mentioned in [13], $E1$ and $E2$ are the optimization objectives of existing methods, which assess how the reconstruction complies with the noisy input, but $E3$ and $E4$ are more meaningful for evaluating the correctness of the clean subspace recoveries. Therefore, we pay more attention to the quantitative indices of $E3$ and $E4$. In the tables, the first and second best performances are marked out with bold and underline, respectively.

The performance of each method in the synthetic experiments are summarized in Table 1 and Table 2, respectively. From the tables, we can see that our methods perform better in terms of $E3$ and $E4$ in most cases. Specifically, the application of MoG makes the matrix based MoG LRMF outperform MC ALM, and the tensor based MoG WLRTF outperform WLRTF. This validates that MoG is able to model a wider range of noise distributions as a universal approximator. Besides, the superiority of MoG WLRTF over MoG LRMF indicates that it better preserves the individual structure of the tensor data.

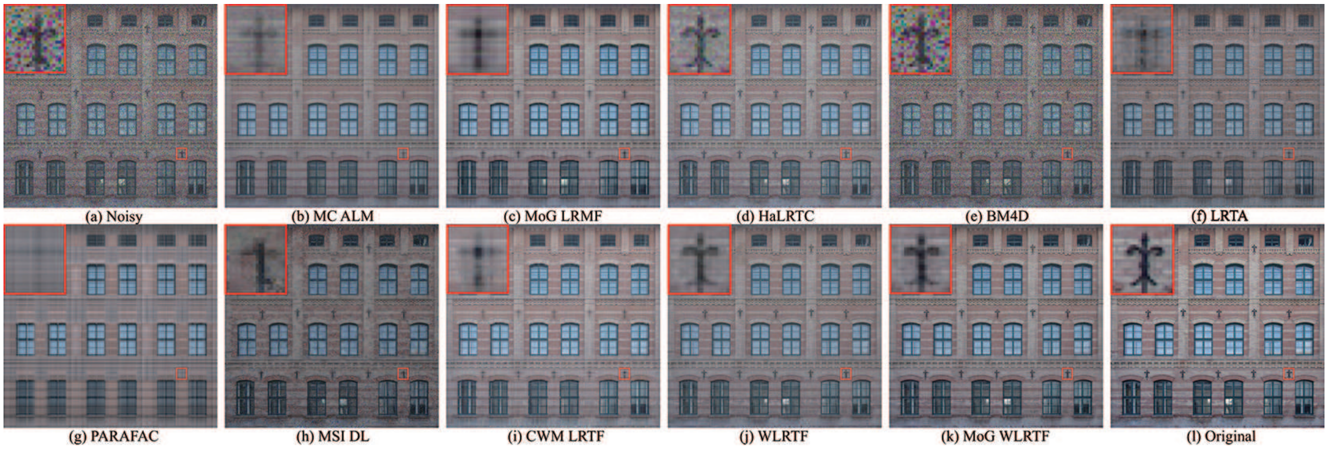

(a) Noisy
(b) MC ALM
(c) MoG LRMF
(d) HaLRTC
(e) BM4D
(f) LRTA
(g) PARAFAC
(h) MSI DL
(i) CWM LRTF
(j) WLRTF
(k) MoG WLRTF
(l) Original

Figure 2. Facade with small mixture noise. (a) Noisy image. (b)-(k) Restored images obtained by competing methods. (l) Original image.

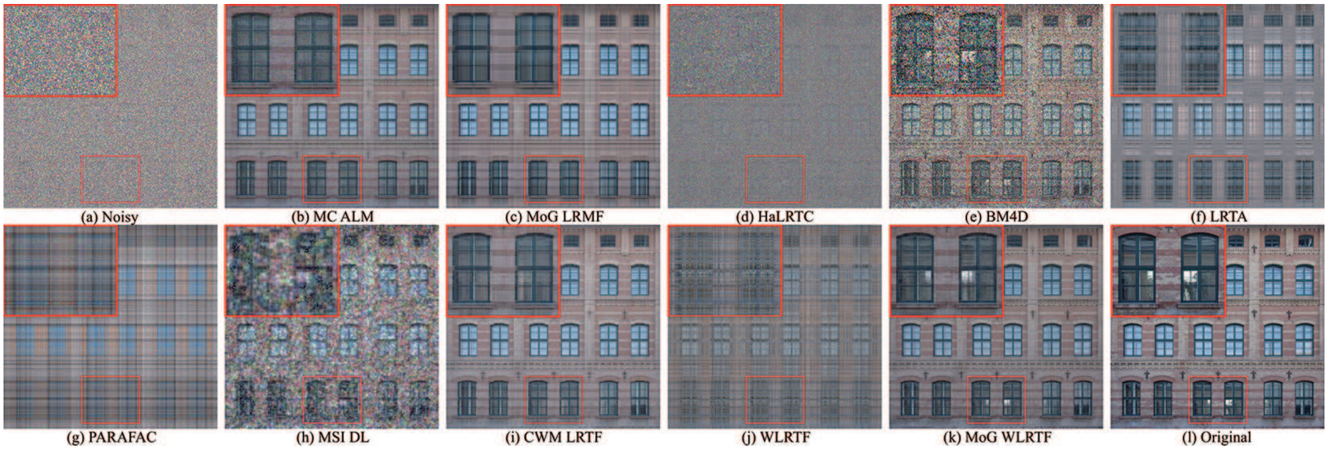

(a) Noisy
(b) MC ALM
(c) MoG LRMF
(d) HaLRTC
(e) BM4D
(f) LRTA
(g) PARAFAC
(h) MSI DL
(i) CWM LRTF
(j) WLRTF
(k) MoG WLRTF
(l) Original

Figure 3. Facade with large mixture noise. (a) Noisy image. (b)-(k) Restored images obtained by competing methods. (l) Original image.

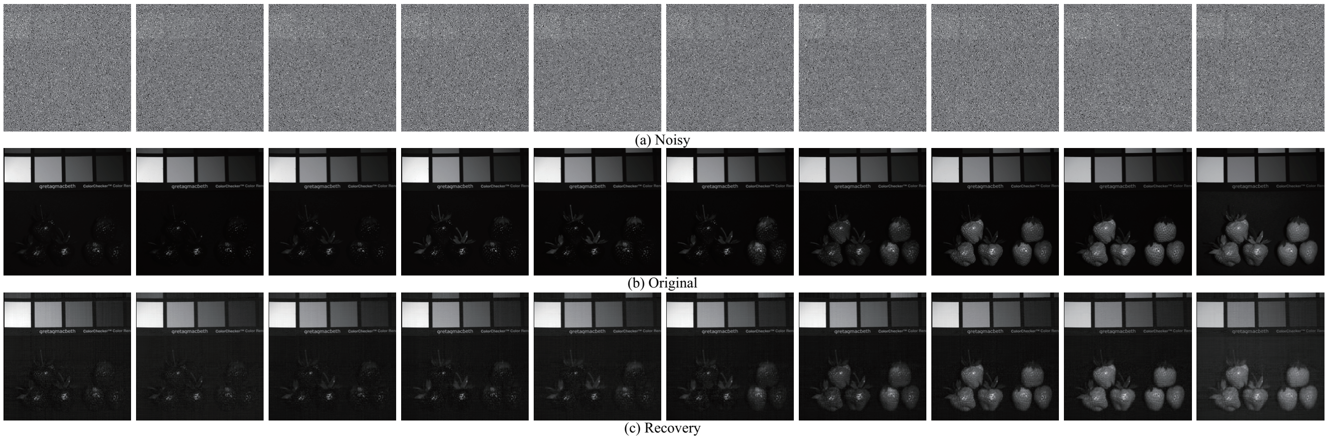

(a) Noisy
(b) Original
(c) Recovery

Figure 4. Ten randomly selected bands of strawberries. (a) Noisy bands. (b) Original bands. (c) Bands recovered by MoG WLRTF.

Table 3. Facade reconstruction performance of competing methods with mixture noise.

Facade		MC ALM	MoG LRMF	HaLRTC	BM4D	LRTA	PARAFAC	MSI DL	CWM LRTF	WLRTF	MoG WLRTF
small scale noise	PSNR	24.61	24.34	23.43	12.00	13.59	13.37	13.53	24.80	24.73	**25.65**
	RSE	0.1133	0.1169	0.1298	0.4838	0.4026	0.4129	0.4062	0.1109	0.1118	**0.1005**
	FSIM	0.9091	0.8954	0.9407	0.8371	0.8318	0.7402	0.8258	0.9435	0.9473	**0.9539**
large scale noise	PSNR	22.09	22.18	14.20	9.204	18.51	16.95	16.71	22.82	17.14	**23.69**
	RSE	0.1515	0.1499	0.3755	0.6673	0.2287	0.2737	0.2817	0.1393	0.2679	**0.1260**
	FSIM	0.8627	0.8525	0.6003	0.7619	0.7667	0.7101	0.7310	0.9117	0.7033	**0.9268**

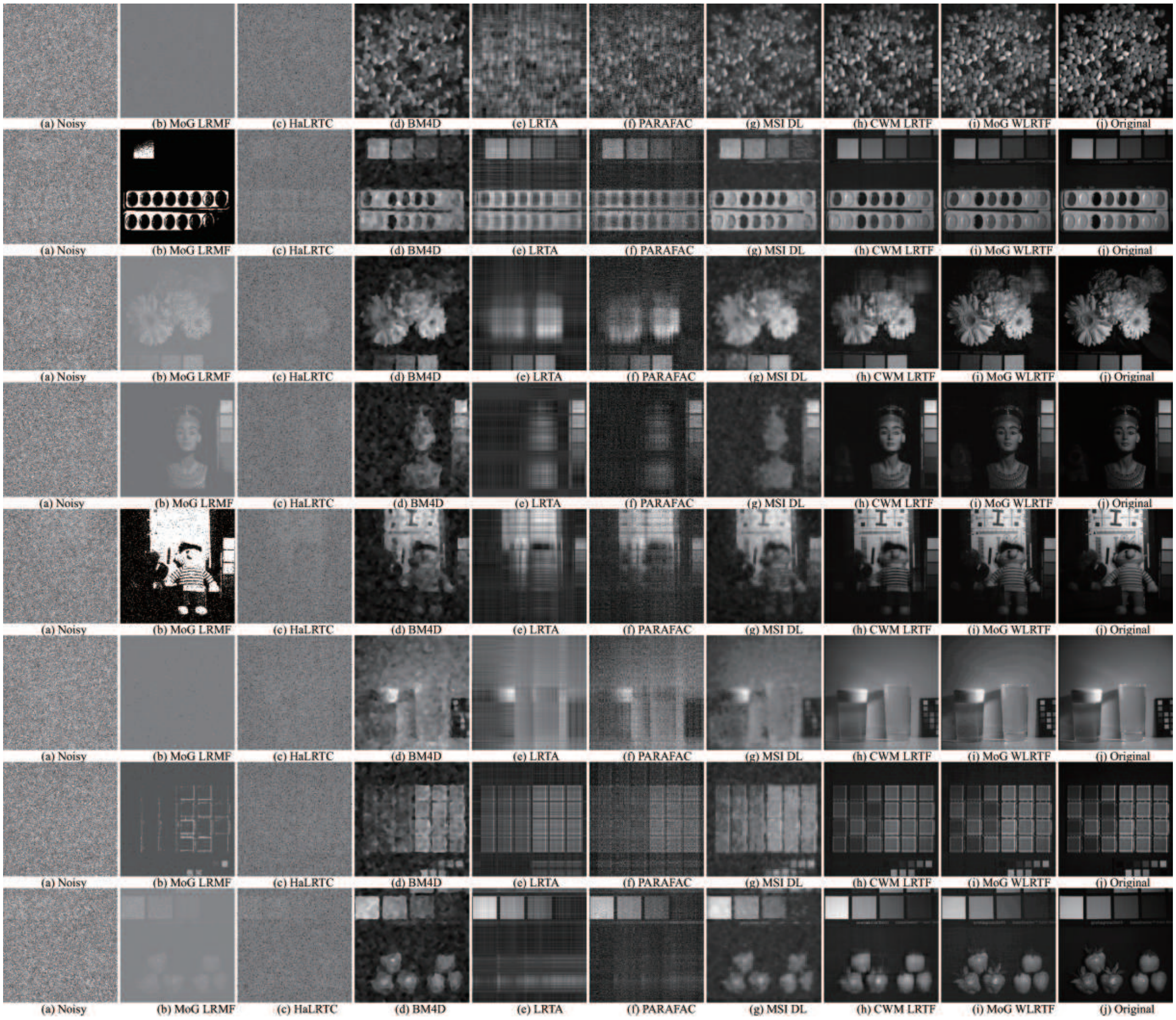


Figure 5. The 31st band of multispectral images. (a) Noisy band. (b)-(i) Restored bands obtained by competing methods. (l) Original band.

3.2. Real image restoration

The images used in this section to evaluate the performance of the competing methods in image restoration are chosen as follows: (1) the benchmark image: the colorful building facade image; (2) a well-known data set: Columbia Multispectral Image Database [27][1]; (3) real hyperspectral image: a HYDICE urban image[2]. Note that each real image used here can be viewed as a 3-order tensor.

Three quantitative image quality indices are adopted to evaluate the performance of each method: peak signal-to-noise ratio (PSNR), relative standard error (RSE) and feature similarity (FSIM) [28]. Larger values of PSNR and FSIM and smaller values of RSE mean a better restoration results.

Simulated image restoration. Firstly, the facade image is randomly sampled with 20% missing entries and added with a relative small scale mixture noise: 20% of the non-missing pixels with the uniformly distribution over $[-35, 35]$, 20% of the rest non-missing pixels with Gaussian noise $\mathcal{N}(0, 20)$ and the rest with another uniformly distribution $\mathcal{N}(0, 10)$. Both the visual and the quantitative results are demonstrated in Figure 2 and Table 3 (the upper row). For better visual comparison, we have also provided a zoom-in version of a local region in Figure 2. It demonstrates that our method performs better in details than the other competing methods when the mixture noise is not very

[1] http://www1.cs.columbia.edu/CAVE/databases/multispectral

[2] http://www.tec.army.mil/hypercube

Table 4. Multispectral image restoration performance of competing methods with mixture noise.

		MoG LRMF	HaLRTC	BM4D	LRTA	PARAFAC	MSI DL	CWM LRTF	MoG WLRTF
Jelly beans	PSNR	8.559	8.444	20.09	16.51	15.84	18.47	19.94	**22.17**
	RSE	1.749	1.773	0.4637	0.7003	0.7565	0.5588	0.4720	**0.3652**
	FSIM	0.5450	0.5312	0.7778	0.7025	0.6487	0.8213	0.8506	**0.8864**
Paints	PSNR	6.342	8.725	20.96	18.76	16.26	19.20	21.75	**27.29**
	RSE	1.757	1.336	0.3267	0.4209	0.5607	0.3998	0.2983	**0.1576**
	FSIM	0.4997	0.4750	0.8016	0.7827	0.6367	0.8182	0.9165	**0.9514**
Flowers	PSNR	9.032	7.480	21.65	18.40	16.54	19.15	22.72	**26.17**
	RSE	2.110	2.522	0.4937	0.7173	0.8888	0.6583	0.4364	**0.2933**
	FSIM	0.7690	0.4172	0.7833	0.8073	0.5271	0.8220	0.9126	**0.9153**
Egyptian statue	PSNR	9.187	7.109	22.04	18.59	17.01	19.43	23.20	**25.22**
	RSE	3.028	3.847	0.6898	1.026	1.231	0.9308	0.6032	**0.4779**
	FSIM	0.8538	0.3696	0.7861	0.8245	0.4243	0.8142	0.9187	**0.9439**
Chart and stuffed toy	PSNR	4.728	7.911	20.77	18.29	16.23	18.95	21.46	**23.95**
	RSE	2.003	1.388	0.3160	0.4201	0.5327	0.3895	0.2919	**0.2191**
	FSIM	0.7303	0.4534	0.7857	0.7729	0.5366	0.8092	0.8901	**0.9221**
Beers	PSNR	11.88	11.56	23.65	19.92	17.14	21.33	21.92	**25.07**
	RSE	0.7858	0.8150	0.2027	0.3114	0.4287	0.2646	0.2474	**0.1722**
	FSIM	0.7875	0.4155	0.8035	0.7535	0.4423	0.8214	**0.9430**	0.9200
Glass tiles	PSNR	8.782	8.380	20.90	18.74	16.49	18.89	21.66	**26.66**
	RSE	1.933	2.025	0.4793	0.6140	0.7961	0.6038	0.4389	**0.2467**
	FSIM	0.5692	0.4690	0.7267	0.7726	0.5738	0.7287	0.9075	**0.9475**
Strawberries	PSNR	8.068	7.762	22.10	19.22	16.78	19.54	18.99	**24.80**
	RSE	2.074	2.149	0.4123	0.5745	0.7607	0.5534	0.5900	**0.3021**
	FSIM	0.7474	0.3997	0.7721	0.8059	0.4932	0.8128	0.9229	**0.9283**

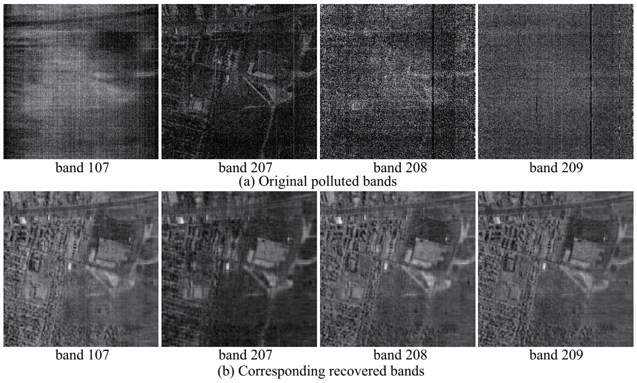
large.

Secondly, in order to further compare the reconstruction ability of each method, we add a larger mixture noise to the facade and multispectral images. Each image is resized to half for all channels/bands and rescaled to [0,1]. The larger mixture noise are added as in the synthetic experiments: 20% missing entries, 20% of the non-missing pixels with the uniformly distribution over $[-5, 5]$, 20% of the rest non-missing pixels with Gaussian noise $\mathcal{N}(0, 0.2)$ and the rest with another uniformly distribution $\mathcal{N}(0, 0.01)$.

The facade reconstruction results are shown in Figure 3 and Table 3 (the lower row). In Figure 3, we also show a zoom-in version of a local region for comparison. We can see that our MoG WLRTF is more robust to larger mixture noise than other methods.

For better visual demonstration of the multispectral image restoration result, we randomly choose ten selected bands of strawberries as shown in Figure 4. Meanwhile, we select the 31st band of these multispectral images to show our restoration results compared with other competing methods as shown in Figure 5 and Table 4. The superiority of the proposed MoG WLRTF method can be observed in multispectral image restoration.

Real Hyperspectral image restoration. Here we use a HYDICE urban image for demonstration. This real hyperspectral image contains several bands seriously polluted by the atmosphere and water absorption, and traditional methods generally discarded these seriously polluted bands before processing [29]. Figure 6 shows the restoration results of four seriously polluted bands in HYDICE urban image. It can be observed that MoG WLRTF can still have a good performance in dealing with such real gross noise.



(a) Original polluted bands

(b) Corresponding recovered bands

Figure 6. Real hyperspectral image restoration. (a) Original polluted bands. (b) Corresponding bands recovered by MoG WLRTF.

4. Conclusion

In this paper, we propose a new MoG based weighted low-rank tensor factorization method to estimate subspaces from high-dimensional data which are disturbed by noises with a complex distribution. Compared with the existing matrix methods, which lose the salient structure of the individual data, our method is capable of better preserving this information and performing better when the data are polluted with a large percentage. Additionally, our method also performs better than other tensor methods which are just optimal for Gaussian or Laplace noise. Both synthetic experiments and the real applications demonstrate the effectiveness of our method under complex noisy tensor data.

Acknowledgment

We thank Lin Lin for the helpful discussions on experiments. This work was supported by the National Natural Science Foundation of China (Grant No. 61303168, 61333019, 11501440, 61373114).

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[2] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scal ing via an n-way generalization of "eckart-young decomposition. *Psychometrika*, 35:283–319, 1970.

[3] E. C. Chi and T. G. Kolda. Making tensor factorizations robust to non-gaussian noise. *arXiv preprint arXiv:1010.3043*, 2013.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[5] K. J. Friston, S. Williams, R. Howard, R. S. Frackowiak, and R. Turner. Movement-related effects in fmri time-series. *Magnetic Resonance in Medicine*, 35:346–355, 1996.

[6] H. Huang and C. Ding. Robust tensor factorization using r1-norm. In *Proceedings of the IEEE international conference on computer vision*, pages 1–8, 2008.

[7] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[8] L. Li, W. Huang, I.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *Image Processing, IEEE Transactions on*, 13(11):1459–1472, 2004.

[9] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[10] J. Liu, P. Musialski, P. Wonka, and J. P. Ye. Tensor completion for estimating missing values in visual data. *IEEE TPAMI*, 34(1):208–220, 2013.

[11] X. Liu, S. Bourennane, and C. Fossati. Denoising of hyperspectral images using the parafac model and statistical performance analysis. *Geoscience and Remote Sensing, IEEE Transactions on*, 50(10):3717–3724, 2012.

[12] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *Image Processing, IEEE Transactions on*, 22(1):119–133, 2013.

[13] D. Meng and F. D. L. Torre. Robust matrix factorization with unknown noise. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1337–1344. IEEE, 2013.

[14] D. Meng, B. Zhang, Z. Xu, L. Zhang, and C. Gao. Robust low-rank tensor factorization by cyclic weighted median. *Science China Information Sciences*, 58(5):1–11, 2015.

[15] H. Murase and S. K. Nayar. Learning and recognition of 3d objects from appearance. In *Qualitative Vision, 1993., Proceedings of IEEE Workshop on*, pages 39–50. IEEE, 1993.

[16] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang. Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2949–2956. IEEE, 2014.

[17] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin. Scalable bayesian low-rank decomposition of incomplete multiway tensors. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1800–1808, 2014.

[18] N. Renard, S. Bourennane, and J. Blanc-Talon. Denoising and dimensionality reduction using multilinear tools for hyperspectral images. *Geoscience and Remote Sensing Letters, IEEE*, 5(2):138–142, 2008.

[19] P. Scheunders and S. De Backer. Wavelet denoising of multicomponent images using gaussian scale mixture models and a noise-free image as priors. *Image Processing, IEEE Transactions on*, 16(7):1865–1872, 2007.

[20] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *JOSA A*, 4(3):519–524, 1987.

[21] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[22] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[23] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using powerfactorization and g-pca. *International Journal of Computer Vision*, 79(1):85–105, 2008.

[24] M. J. Wainwright and E. P. Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In *NIPS*, pages 855–861. Citeseer, 1999.

[25] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[26] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, volume 10, pages 211–222. SIAM, 2010.

[27] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *Image Processing, IEEE Transactions on*, 19(9):2241–2253, 2010.

[28] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: a feature similarity index for image quality assessment. *Image Processing, IEEE Transactions on*, 20(8):2378–2386, 2011.

[29] Q. Zhao, D. Meng, X. Kong, Q. Xie, W. Cao, Y. Wang, and Z. Xu. A novel sparsity measure for tensor recovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 271–279, 2015.

[30] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang. Robust principal component analysis with complex noise. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 55–63, 2014.

[31] Q. Zhao, L. Zhang, and A. Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(9):1751–1763, 2015.